

Actas de las XIX Jenui. Castellón, 10-12 de julio 2013
ISBN: 978-84-695-8051-6 DOI: 10.6035/e-TiIT.2013.13
Páginas: 335-341

PeLP: Plataforma para el Aprendizaje de Lenguajes de Programación

Xavier Baró, David Masip, Elena Planas y Julià Minguillón
Estudis d'Informàtica, Multimèdia i Telecomunicació
Universitat Oberta de Catalunya
Rambla del Poblenou 156, 08018 Barcelona
{xbaro, dmasipr, eplanash, jminguillona}@uoc.edu

Resumen

En este artículo se presenta una plataforma para el aprendizaje en línea de lenguajes de programación. Su principal objetivo es ofrecer una retroalimentación inmediata a los estudiantes en sus ejercicios de programación. Además, la plataforma contiene un conjunto de servicios web que permiten la compilación y verificación de código en línea. La plataforma no sólo facilita a los docentes la gestión y corrección de las actividades de los estudiantes y les da un retorno inmediato, sino que abre la puerta a la realización de aplicaciones web que permitan la resolución en línea de ejercicios de programación sin la necesidad de instalar ningún entorno de desarrollo ni depender de ninguna plataforma concreta. Actualmente se está implantando en asignaturas de los Grados en Ingeniería Informática y Tecnologías de la Telecomunicación.

Abstract

This paper presents an eLearning platform for programming languages. Its main aim is to provide students with immediate feedback on their programming exercises. In addition, the platform contains a set of web services that allow online compilation and code verification. The platform does not only simplify the management and correction of activities to teachers and gives immediate feedback to students, but also opens the door to the realization of web applications that allow online resolution of programming exercises without installing any development environment and totally platform independent. The platform is currently on implantation process for Computer Science and Telecommunication Degrees.

Palabras clave

Recursos educativos, aprendizaje en línea, lenguajes de programación.

1. Introducción

Aprender a programar en un determinado lenguaje de programación suele conllevar el aprendizaje no sólo de las estructuras y singularidades del lenguaje, sino también de un entorno de programación y un conjunto de herramientas específicas para su ejecución. Para una persona que ya tenga nociones de programación, este aprendizaje se convierte en localizar las equivalencias con sus conocimientos previos, y suele ser más asequible. El problema cambia considerablemente cuando hablamos de estudiantes de primer año de Grado, que generalmente no han programado nunca y que en la mayoría de casos desconocen qué significa programar. El aprendizaje conjunto de la programación, un determinado lenguaje y las herramientas necesarias resultan en una curva de aprendizaje muy pronunciada.

Para suavizar dicha curva de aprendizaje, una posibilidad es retardar la necesidad de las herramientas hasta que los estudiantes tengan las nociones básicas sobre el lenguaje de programación. De este modo, el estudiante ya conoce previamente el resultado esperado de su código y puede centrarse en cómo utilizar las herramientas convencionales para el lenguaje objetivo. Actualmente existen herramientas que pueden ayudar en este sentido, por ejemplo algunos correctores de ejercicios de programación [1, 2, 3, 4], que permiten valorar la calidad de un ejercicio de codificación y compiladores en línea de código [5], que permiten probar pequeños códigos. En el caso de los sistemas de corrección analizados, generalmente basados o derivados de *Ceilidh* [6], se ha observado que en general han dejado de ser mantenidos de forma abierta y su instalación e integración en plataformas abiertas es muy costosa. Uno de los entornos más utilizados y que se mantiene abierto es Mooshak [7], que presenta unas características similares a nuestra plataforma. A diferencia de Mooshak PeLP, permite integración con los servicios de información de la universidad, y ofrece además una capa de servicios Web que permite llamadas directas desde cualquier aplicación, sin necesidad

de realizar envíos de código mediante formularios. En el caso de los sistemas de compilación en línea, la mayoría de los sistemas analizados no tienen el código abierto y no permiten ser integrados en sistemas propios.

Dada la complejidad y coste en la adaptación de los sistemas disponibles, pero teniendo en cuenta los potenciales beneficios de este tipo de herramientas en el aprendizaje de la programación, se decidió desarrollar una plataforma propia. La plataforma que se presenta en este artículo tiene como finalidad dar soporte a los estudiantes de primer año de los Grados en Ingeniería Informática y Tecnologías de la Telecomunicación en sus ejercicios de programación. Uno de los puntos fuertes de la herramienta es la retroalimentación inmediata que ofrece a los estudiantes, permitiéndoles autoevaluarse a lo largo del curso. También se ha tenido en cuenta la posibilidad que la plataforma pueda integrarse fácilmente en distintas instituciones.

El resto del artículo se organiza de la siguiente forma: en la Sección 2 se describen los objetivos, arquitectura y funcionalidades de la plataforma; en la Sección 3 se explica su implantación en el entorno de la *Universitat Oberta de Catalunya (UOC)*, el impacto esperado y el piloto desarrollado en la asignatura de Fundamentos de Programación; finalmente, la Sección 4 presenta las conclusiones y el trabajo futuro.

2. Descripción de la plataforma

En esta sección se describen los antecedentes de la plataforma PeLP, se exponen sus objetivos y se detalla su arquitectura y funcionalidades.

2.1. Antecedentes

La plataforma PeLP tiene sus orígenes en el denominado *corrector automático*, una herramienta web basada en PHP que permite ejecutar un fichero de código C utilizando un conjunto de juegos de pruebas básicos, y devolviendo una retroalimentación inmediata. Esta herramienta se implantó el curso 2004-2005 en la asignatura Fundamentos de Programación de la *UOC*. Hasta la fecha, un total de 4,700 estudiantes de las Ingenierías Técnicas de Informática, 840 de la Ingeniería Técnica en Telecomunicaciones, así como 820 y 230 estudiantes de los recientes Grados en Ingeniería Informática y de Tecnologías de la Telecomunicación respectivamente han utilizado esta herramienta para entregar sus ejercicios de programación.

En estos años, aunque ha demostrado ser una herramienta muy útil tanto para los estudiantes como para los profesores, también se han evidenciado sus limitaciones. Por ejemplo, el corrector sólo acepta código C contenido en un único fichero, lo que limita el tipo

de ejercicios que se pueden validar. Así mismo, tiene una gestión muy simple de las entregas, que no permite que un profesor pueda acceder de forma fácil a las entregas de sus estudiantes, y además no permite a los estudiantes mantener un registro de las mismas.

2.2. Objetivos

Partiendo de la experiencia acumulada a partir del uso del *corrector automático*, los problemas encontrados durante su tiempo de funcionamiento y el deseo de mejorar los recursos docentes a disposición de los estudiantes, se ha concebido la plataforma PeLP. Esta nueva plataforma está pensada para proporcionar las capacidades de *compilación online* y *corrección automática* de ejercicios de codificación, siendo diseñada para permitir la creación de nuevos recursos interactivos que requieran el análisis de un código de entrada. Las bases sobre las que se ha diseñado la plataforma que se presenta son:

Seguridad: Dado que una de las posibilidades de la plataforma es poder ser utilizada para la entrega de actividades de programación, garantizar el acceso controlado a las actividades entregadas ha sido una de las prioridades. En este sentido, se han definido restricciones a varios niveles para asegurar que nadie puede acceder a lo que no debería.

Integración: Para facilitar la gestión de los usuarios de la plataforma, se ha trabajado en todo momento teniendo en mente la posibilidad de utilizar los métodos de autenticación de los usuarios en los entornos de la propia universidad. La plataforma se ha diseñado para que permita utilizar un sistema externo de gestión de usuarios y asignaturas.

Escalabilidad: Teniendo en cuenta la gran cantidad de lenguajes de programación disponibles y que la elección de un determinado lenguaje puede depender de muchos factores, la plataforma se ha diseñado para facilitar la introducción de nuevos lenguajes. En la versión actual, existen implementaciones tanto para C como para Java.

Apertura: En todo momento se ha percibido la plataforma como una herramienta abierta. Por un lado, evitando limitar la plataforma a los requisitos concretos de la universidad donde se está implantando. También se ha definido una capa de servicios de acceso a la plataforma, que permite la utilización de la misma desde cualquier otra aplicación, facilitando tareas de compilación y ejecución en la nube, sin necesidad de instalar ningún entorno de programación. Finalmente, todo el código se ha publicado en *GitHub*¹ con una licencia abierta, facilitando que quien quiera pueda utilizarlo y adaptarlo a sus necesidades.

¹GitHub www.github.com

Internacionalización: En todo momento se ha pretendido que la plataforma sea multilingüe. Actualmente está disponible en catalán, castellano e inglés.

2.3. Arquitectura

En esta sección se detallan las características principales de la plataforma PeLP y de los módulos que la conforman. En la Figura 1 se puede ver el conjunto de capas en que se organiza la plataforma. El núcleo de la plataforma es el motor, el cual gestiona todos los flujos de datos entre los distintos módulos de la aplicación. Una de las tareas más delicadas del motor es asegurar que nadie pueda acceder a ningún recurso al que no tiene derecho a acceder. Se han definido los siguientes módulos:

Gestor de Actividades: Gestiona las actividades disponibles para los estudiantes. Cada asignatura puede tener varias actividades en un semestre, las cuales pueden estar delimitadas en una ventana temporal o no. Cada actividad puede tener asociado un conjunto de juegos de prueba públicos y/o privados. En el caso de los públicos, los estudiantes pueden ver en qué consisten; en el caso de los privados, sólo saben si lo han superado o no.

Gestor de Entregas: Se encarga de acceder a las entregas realizadas por los estudiantes a las distintas actividades. Para cada entrega se guardan tanto los ficheros que la componen como el resultado obtenido del análisis del código.

Módulo de Análisis de Código: Este módulo se encarga de analizar los ficheros de código de una entrega. Dado que el análisis depende del lenguaje de programación utilizado, este módulo define la operativa básica, y se requiere un adaptador para cada lenguaje. Actualmente el análisis del código se limita a la compilación y las pruebas de ejecución definidas por los juegos de prueba, así como el tiempo de ejecución para cada una de ellas. Dado que los códigos pueden contener errores, la ejecución se hace en un entorno protegido y con limitaciones de recursos y temporales.

Conexión al Campus: Este módulo es sin lugar a dudas el más delicado para la integración de la plataforma en cualquier entorno educativo. Él es el responsable de autenticar al usuario y aportar toda la información referente a asignaturas, estudiantes, profesores, etc. El módulo define las interfaces que cubren las necesidades de la plataforma, y su implementación se delega a una implementación que tenga en cuenta los requisitos de cada universidad/entidad. En la versión actual, se ha desarrollado un adaptador que no requiere comunicación externa, sino que proporciona informa-

ción local, muy útil para entornos pequeños o para pruebas. También se ha desarrollado un adaptador basado en OAuth2² para conectar con el campus de la UOC.

Módulo de Persistencia: La función de este módulo es que los datos que se generan en la plataforma queden correctamente guardados en disco y en la base de datos, y que se puedan recuperar cuando se precise. Entre los datos que se gestionan están toda la información de configuración de la plataforma, las entregas y los resultados de las entregas.

Además de las capas y módulos que dan vida a la plataforma, se ha definido una *Capa de Servicios*, la cual permite acceder a las funcionalidades de la plataforma mediante servicios web. El acceso mediante servicios web facilita poder utilizar las funcionalidades de análisis de código desde aplicaciones externas, como por ejemplo páginas web o aplicaciones en dispositivos móviles.

2.4. Funcionalidades

La primera prioridad de la plataforma es sustituir al actual *corrector automático*, por lo que las primeras funcionalidades que se han activado en la plataforma son todas aquellas relacionadas con la gestión de actividades y entregas de ejercicios por parte de los estudiantes. En concreto, se ha creado un cliente web para que los estudiantes puedan acceder a las actividades y entregar sus ejercicios. En la Figura 2 se muestra la pantalla inicial del cliente. Esta pantalla permite subir ficheros de código a la plataforma para que sean compilados. Esta funcionalidad se ha dejado abierta a cualquier usuario, por lo que está accesible sin necesidad de autenticarse.

La autenticación en la plataforma se puede hacer mediante usuario y contraseña o bien utilizando un servicio externo. Por ejemplo, en la Figura 2 podemos ver que se da la opción de utilizar el sistema de la UOC. Cuando un usuario está autenticado correctamente, se muestra su información y se le permite elegir una asignatura en la que esté matriculado, el aula y finalmente una actividad. Esta selección es igual para estudiantes y profesores, con la diferencia que un estudiante sólo puede tener un aula, mientras que un profesor puede tener varias.

Cuando un usuario elige una actividad concreta, le aparece una pestaña adicional que le permite ver un registro de entregas. En caso que sea un estudiante, en esta pestaña se le muestran las entregas que ha efectuado para dicha actividad, con un resumen de los resultados obtenidos (ver Figura 3). En caso que el usuario sea un profesor, se muestra la misma información para

²OAuth2 <http://oauth.net/2/>

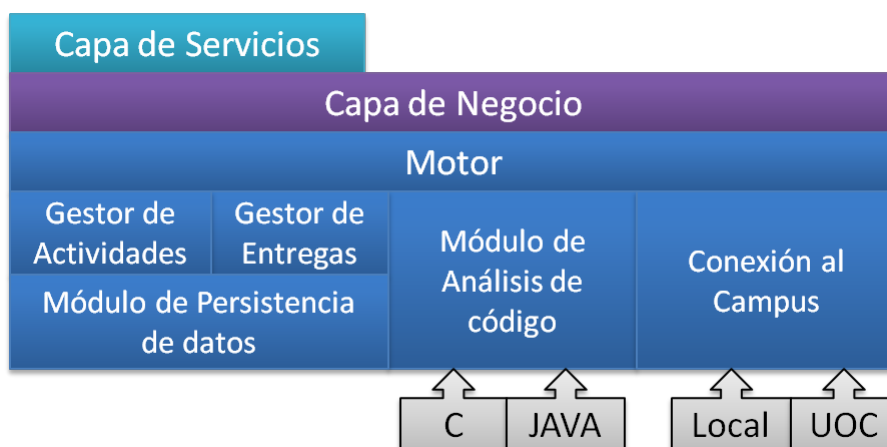


Figura 1: Estructura de capas que conforman la plataforma PeLP.

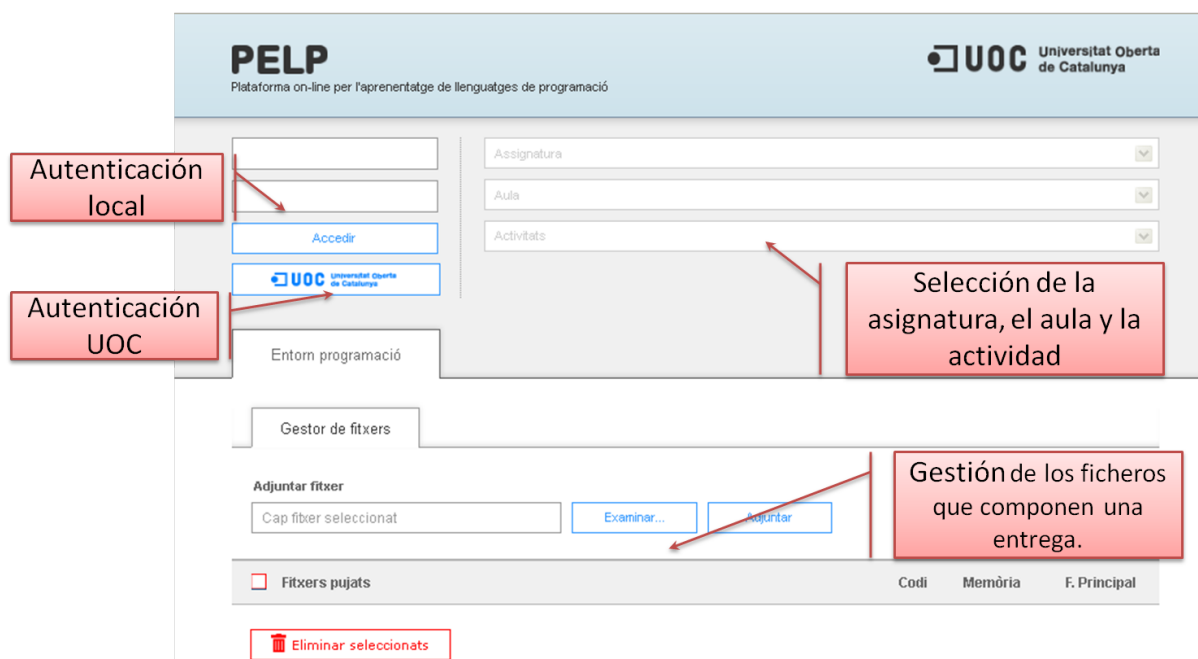


Figura 2: Página principal del cliente web para la UOC.

Entrega	Data	Intents	Compilació	Tests públics	Tests privats
> user3	11/02/2013	0	✓	0 0	0 0
> user4					
> Entrega 3	16/01/2013		✗	0 0	0 0
> Entrega 4	14/02/2013		✓	0 0	0 0
Fitxer			Codi	Memòria	F. Principal
test.java			✓		✓
Tests públics					
Tests privats					
> user6	16/01/2013	0	✓	0 0	0 0

Figura 3: Registro de entregas.

el último registro de cada estudiante de su aula. En ambos casos, al seleccionar un registro, se puede acceder a todo el detalle asociado.

En la experiencia previa con el *corrector automático* se detectó que los estudiantes utilizaban el corrector para programar mediante ensayo y error. Es decir, iban mandando su código con pequeñas modificaciones para ver si compilaba y si pasaba los tests. Esta forma de trabajar, además del coste de recursos que conlleva para el servidor, creemos que no es la adecuada, pues los estudiantes no reflexionaban sobre los motivos de sus errores. Para evitar esta forma de trabajar, cada actividad lleva asociada un número máximo de entregas. De este modo, el estudiante necesita utilizar sus herramientas y analizar los resultados obtenidos antes de volver a mandarla a la plataforma de corrección.

2.5. Portabilidad

Todo el código de la plataforma se ha publicado íntegro en GitHub (<https://github.com/xbaro/PeLP>), permitiendo a cualquier persona que lo desee hacer una copia de la versión actual y evolucionarla o adaptarla a sus necesidades. Si nos fijamos en la Figura 1, podemos observar que tanto los módulos de análisis de código como la conexión al campus aparecen como figuras externas a la plataforma. El motivo es que se han definido como módulos de extensión de la plataforma, lo que facilita incluir nuevos lenguajes de programación y nuevos módulos de conexión que

se adapten a cada institución.

En ambos casos, la extensión se realiza mediante la implementación de una interfaz JAVA, que define qué métodos se deben definir en cada implementación. En el caso de añadir un nuevo lenguaje de programación, se deberá indicar cómo se compila y cómo se ejecuta, lo que permitirá a la plataforma poder obtener resultados de compilación y de ejecución, que serán gestionados internamente. En el caso de la conexión con otro campus, los métodos que hay que implementar están relacionados con la autenticación, obtener las aulas de un usuario, el idioma preferido del usuario en caso de campus con varios idiomas, etc.

3. Prueba piloto en el contexto de la UOC

La plataforma PeLP se ha diseñado dentro de UOC, siendo las asignaturas de programación básicas las más beneficiadas.

En una primera fase, la plataforma PeLP se implantará en la asignatura Fundamentos de Programación. Esta asignatura, con unos 250 estudiantes matriculados por semestre, consta de dos prácticas desarrolladas en lenguaje C. Utilizando la nueva plataforma PeLP, el estudiante podrá enviar su código mediante la interfaz habilitada para su evaluación, y recibirá los resultados obtenidos sobre un conjunto de juegos de prueba (tanto públicos como privados). De este modo, el estu-

diente recibirá una retroalimentación inmediata sobre posibles errores en el proceso de compilación y ejecución, y tendrá en todo momento constancia de qué funcionalidad no está correctamente implementada.

En una segunda fase, se plantea extender el uso de PeLP al resto de asignaturas de programación: Prácticas de Programación, Diseño y Programación Orientada a Objetos y el Taller de Java para estudiantes con conocimientos previos de programación. Estas asignaturas están vinculadas a los primeros cursos del Grado en Ingeniería Informática, y se imparten también en el Grado de Tecnologías de la Telecomunicación de la UOC. Actualmente, promedian globalmente más de 600 estudiantes por semestre.

Además, al tratarse de asignaturas de primer curso y con una carga de trabajo considerable, resulta bastante habitual que se encuentren entre las asignaturas críticas a nivel de rendimiento académico.

El sistema de entrega y corrección que proporciona la plataforma PeLP proporciona diversas ventajas, especialmente en un entorno asíncrono como es el modelo educativo de la UOC (no presencial). En este entorno, aunque el estudiante está acompañado durante todo el curso por un profesor que tutoriza su aprendizaje y le proporciona periódicamente una retroalimentación personalizada, es necesario disponer de herramientas de soporte adicional para mejorar el proceso de aprendizaje. En este sentido, la plataforma PeLP da soporte directo a los estudiantes de las asignaturas de programación durante la elaboración de sus ejercicios de codificación, proporcionándoles un conjunto de ejercicios de autoevaluación, una retroalimentación inmediata y un sistema ágil para realizar las entregas requeridas en la asignatura.

Además de las ventajas en el proceso de aprendizaje del estudiante, la plataforma tiene otro impacto positivo en la evaluación de las prácticas por parte del equipo docente. Pasada la fecha límite, los profesores pueden acceder a un resumen de los resultados obtenidos por cada estudiante, teniendo un acceso rápido e inmediato al código proporcionado. Los resultados de compilación y de ejecución sobre los juegos de prueba públicos y privados sirven para validar el correcto funcionamiento del código, por lo que los docentes pueden dedicar el tiempo de corrección a analizar otros aspectos sobre la estructura, modularidad y estilo de código, reduciendo el coste de corrección y dando una retroalimentación personalizada a los estudiantes más detallada.

4. Conclusiones y trabajo futuro

La plataforma presentada es un avance respecto al sistema actual, diseñada para poder ser adaptada de forma fácil a distintas instituciones y necesidades. La

primera fase de la plataforma se ha centrado en cubrir los aspectos de entrega y corrección de código y los servicios para acceder a la plataforma, abriendo el camino a nuevas aplicaciones que puedan explotarla.

Actualmente se está trabajando en ampliar el tipo de pruebas que se pueden aplicar a los programas, así como la posibilidad de añadir librerías o ficheros adicionales a las entregas en el paso de compilación. También se está trabajando en mejorar la interfaz de administración de la plataforma, para facilitar la gestión de asignaturas y tareas. Además está previsto integrar un sistema anti-fraude y ampliar el análisis de los códigos entregados, incluyendo indicadores de calidad (uso de comentarios, sangrías, modularidad, ...).

También se está preparando el inicio de la prueba con estudiantes reales, prevista para Septiembre del 2013. Dado que el número de estudiantes será de alrededor de 250 en la prueba piloto, se están efectuando pruebas de carga en los servidores, para poder prevenir contingencias relacionadas con la configuración de los servidores.

Como trabajo futuro queremos integrar en la plataforma el acceso a recursos relacionados con la programación, tanto aquellos totalmente abiertos como recursos de acceso limitado a los estudiantes de una determinada asignatura.

Agradecimientos

Este trabajo se ha financiado con las ayudas a la innovación docente *APLICA 2012* de la Universitat Oberta de Catalunya. Parte del diseño e implementación lo han llevado a cabo Juan Francisco Sánchez y Juan Antonio Mangas, del equipo de Tecnología Educativa de la UOC.

Referencias

- [1] G. Edmunds, "Experiences using caape: computer assisted assessment of programming exercises," *Computers & Education*, vol. 15, no. 1-3, pp. 45-48, 1990.
- [2] C. Higgins, T. Hegazy, P. Symeonidis, and A. Tsin-tsifas, "The coursemarker cba system: Improvements over ceilidh," *Education and Information Technologies*, vol. 8, pp. 287-304, Sept. 2003.
- [3] P. Longo, A. Sterbini, and M. Temperini, "Tsw: A web-based automatic correction system for c programming exercises," in *Proceedings of the 2nd World Summit on the Knowledge Society: Visioning and Engineering the Knowledge Society. A Web Science Perspective*, WSKS '09, (Berlin, Heidelberg), pp. 13-21, Springer-Verlag, 2009.

- [4] C. Douce, D. Livingstone, and J. Orwell, "Automatic test-based assessment of programming: A review," *ACM Transactions on Computing Education / ACM Journal of Educational Resources in Computing*, vol. 5, 2005.
- [5] M. Mohtashim, "Compile Online." <http://www.compileonline.com>. Visitado Febrero 2013.
- [6] S. D. Benford, E. Burke, E. Foxley, N. H. Gutteridge, and Z. A. Mohd, "Ceilidh: A course administration and marking system," in *Proceedings of the 1st International Conference on Computer Based Learning in Science*, (Viena, Austria), 1993.
- [7] J. P. Leal and F. Silva, "Using mooshak as a competitive learning tool," in *The 2008 Competitive Learning Symposium, ACM-ICPC World Finals*, 2008.